



# Open Research Online

---

The Open University's repository of research publications  
and other research outputs

## GAOOLE: a Gaia Design of Agent-Based Online Collaborative Learning Environment

### Conference or Workshop Item

#### How to cite:

Liu, Shuangyan; Joy, Mike and Griffiths, Nathan (2009). GAOOLE: a Gaia Design of Agent-Based Online Collaborative Learning Environment. In: Proceedings of the 8th European Conference on e-Learning, Academic Conferences and Publishing International, pp. 339–350.

For guidance on citations see [FAQs](#).

© 2009 The Authors

Version: Accepted Manuscript

Link(s) to article on publisher's website:

<http://academic-conferences.org/ecel/ecel2009/ecel09-home.htm>

---

Copyright and Moral Rights for the articles on this site are retained by the individual authors and/or other copyright owners. For more information on Open Research Online's data [policy](#) on reuse of materials please consult the policies page.

---

[oro.open.ac.uk](http://oro.open.ac.uk)

# GAOOLE: a Gaia Design of Agent-Based Online Collaborative Learning Environment

Shuangyan Liu, Mike Joy and Nathan Griffiths  
University of Warwick, Coventry, UK

[Shuangyan.Liu@warwick.ac.uk](mailto:Shuangyan.Liu@warwick.ac.uk)

[M.S.Joy@warwick.ac.uk](mailto:M.S.Joy@warwick.ac.uk)

[Nathan.Griffiths@warwick.ac.uk](mailto:Nathan.Griffiths@warwick.ac.uk)

**Abstract:** An intelligent collaborative learning environment (iCLE) provides an online learning community with an interactive and multi-functional work area with intelligent support for the whole cycle of collaborative education, including organizing teams, advising on group work and communication, tutoring, and testing individual contributions. An agent-based approach lends itself to developing iCLE systems since many of the desired properties and requirements of iCLE systems coincide with those provided by the use of agents, such as autonomy, reactivity and proactivity (goal-oriented). Existing agent-based designs for online collaborative learning identify the agent types and the system topology, but lack certain design specifications. In particular, there is a lack of precision with respect to areas such including: the key roles that intelligent agents can play in online collaborative learning management; the computational resources consumed and generated by a role for performing a pedagogical task; the protocols adopted for the interactions between different roles; the agent types with mapped roles and the number of instances of each type in an actual system; and the services that the agents provide. Fully specifying these aspects will enable the system to fully exploit the strengths of agents (including pro-activeness, autonomy and flexibility). In this paper, we propose a new design, GAOOLE (Gaia Design of Agent-based Online Collaborative Learning Environment), which includes a detailed analysis and design specification. It consists of five sub-models: the environment model (describing the computational resources in a collaborative learning environment that are needed by the identified roles and their relationships with them), the roles model (describing the attributes of the roles for managing online collaborative learning – responsibilities, permissions, activities and protocols), the interaction model (defining the protocols for each type of inter-role interaction), the agent model (defining the types of agents and the number of instances of each agent type in actual system) and the services model (describing the services associated with each agent type). We specify these models in this paper by following the Gaia methodology. By applying an established agent-oriented methodology we develop a detailed system design that makes full use of the agent-based approach both in terms of system development, for example by facilitating use of existing components and systems, and in system use, providing characteristics such as flexibility and pro-activeness. In this paper we give an overview of the design, and focus in particular on how the collaborative aspects of the learning environment are supported.

**Keywords:** Intelligent Collaborative Learning Environment; educational agent; design of agent-based systems; gaia methodology; system integration

## 1. Introduction

Building intelligent web-based educational systems is considered to be an important direction of research in the field of e-learning (Hamdi 2007). Much research work, e.g. (Baylor 2002, Lin, et al. 2005, Piramuthu 2005), has shown the significance of utilizing agents for providing intelligence in web-based educational systems.

A collaborative learning environment (CLE) is a web-based educational system that provides collaborative learning specific functionalities (i.e. structuring and managing the collaboration (Reimann 2003)) as well as other supporting functionalities for online learning (e.g. designing, managing and delivering the learning contents). Dimensions along which *collaboration can be structured* include the allocation of members to groups (Cohen 1994), assigning group members to roles such as 'producer' and 'reviewer', regulating who can interact with whom over time (Anderson, et al. 2000), etc. Forms through which *collaboration can be managed* include collecting interaction data, constructing model of interaction, comparing with desired state, moderating (Reimann 2003), etc. The *supporting functionalities* constitute the basic platform for online collaborative learning as for other e-learning forms, which include administration, content management, learning workplace and tools for interaction (e.g. chats, forums, bulletin-boards) (Mencke and Dumke 2007). The collaborative learning specific functionalities can be implemented as different intelligent components of the collaborative learning environment. A convenient metaphor for building the intelligent components is that of educational agents. An educational agent is a software entity that is situated in the learning environment and carries out one or more pedagogical tasks autonomously.

Traditional designs of agent-based systems for online collaborative learning adopt methodologies such as an object-oriented approach for the analysis and design of the system. This is ill-suited to multi-agent systems because of the fundamental mismatch between the abstractions they provide and those that agent-oriented methodologies provide (Wooldridge, et al. 2000). Moreover, a lot of endeavours have been focused on specific technologies to implement agent-based systems for online collaborative learning (e.g. ELMS (Liu, et al. 2006), CITS (Razek, et al. 2002), I-MINDS (Soh, et al. 2006)). In this paper we follow the Gaia methodology (Zambonelli, et al. 2003) to analyze and design an agent-based CLE (i.e. GAOOLE). Such a design is flexible and can be implemented using an appropriate agent programming framework and integrated with existing collaborative learning environments.

The paper is structured as follows. Section 2 introduces a general model of online collaborative learning and the pedagogical tasks that are to be addressed by GAOOLE agents. It also presents the features, structure and limitations of existing activity-based CLEs. In Section 3, we present the results of our Gaia approach to the GAOOLE design, including the models of the environment, roles, interactions, agents and services. Our aim with GAOOLE is to use existing systems where possible, and the integration of GAOOLE agents with existing CLEs is illustrated in Section 4. Finally, Section 5 concludes the paper and identified the ongoing work.

## **2. Background and related work**

### **2.1 A general model of online collaborative learning**

Suh and Lee's general model of online collaborative learning (Suh and Lee 2006) shows the basic process of online collaborative learning: building and arranging a team, developing learning goals and plans, individual learning, team learning, and sharing and evaluating learning outcomes. Their model also addresses the following primary pedagogical tasks to be accomplished by GAOOLE agents, which are associated with the described process.

- Building and arranging a team: advice on team arrangement.
- Developing learning goals and plans: advice on learning goals and plans.
- Individual learning: monitor the individual learning process and give information on current status.
- Team learning: monitor the collaborative learning process and advise group work and communication.
- Sharing and evaluating learning outcomes: analyze the learning process and report on learning outcomes.

### **2.2 Activity-based Collaborative Learning Environment**

The main feature of the existing collaborative learning environments (e.g. Moodle (Moodle.org 2005), LAMS (LAMS International 2002) and Blackboard (Blackboard Inc. 1997)) consists of courses that contain activities and resources. Learners can take an online course by participating in the activities arranged for the course. Here an activity means the work to be completed by learners for the purpose of learning or assessment. There are mainly three common types of activities that are supported by these collaborative learning environments: informative activities (e.g. noticeboard, announcement, and sharing resources), collaborative activities (e.g. chat, forum, and wiki), and assessment activities (e.g. choices, questions and answers, and submitting files).

An activity-based collaborative learning environment is structured mainly for designing, managing and delivering such activities. Such an environment consists of different modules: *content management* allows various activities to be defined and arranged for a particular course; *tools for supporting activities* provide different ways to present the activities; *collaborative workplace* allows online learners to carry out learning activities together and interact with each other remotely in synchronous or asynchronous ways; *administration* allows technicians to maintain the collaborative learning system and course managers to manage online courses.

The existing collaborative learning environments provide those *supporting functionalities* for online collaborative learning. However, they often miss *collaborative learning specific functionalities* which includes collecting interaction data, constructing model of interaction, comparing with desired state of the model, and moderating (Reimann 2003). Several researchers have shown the importance of

these functionalities: such as reducing the cognitive and communication overload of the teachers (Kosba 2004), and positively influencing different aspects of group behaviour (Zumbach, et al. 2006). GAOOLE specifies a multi-agent system for providing those collaborative specific functionalities in the existing collaborative learning environments.

### **3. The Models of GAOOLE**

The Gaia methodology (Wooldridge, et al. 2000, Zambonelli, et al. 2003) has been adopted to analyze and design our multi-agent system for supporting online collaborative learning. By following the Gaia methodology, we can make full use of an agent-oriented approach in terms of system development and system use.

The Gaia is a methodology that guides system developers to define the agent structure (micro-level) and agent organization structure (macro-level) in two phases – analysis and design phase. In the *analysis* phase, the roles model addresses the micro-level aspects, and the interaction model and the environment model address the macro-level aspects. The major tasks are to define a collection of roles of agents, the relationships between agents and between agents and their environments. In the *design* phase, the agent model addresses the macro-level aspects and the services model addresses the micro-level aspects. The major tasks are to establish a mapping from roles to agent types and specify the functions and services provided by individual agent types.

The output of the Gaia process is a collection of practical designable and reusable models. We describe these models of our multi-agent system in detail in the subsequent subsections.

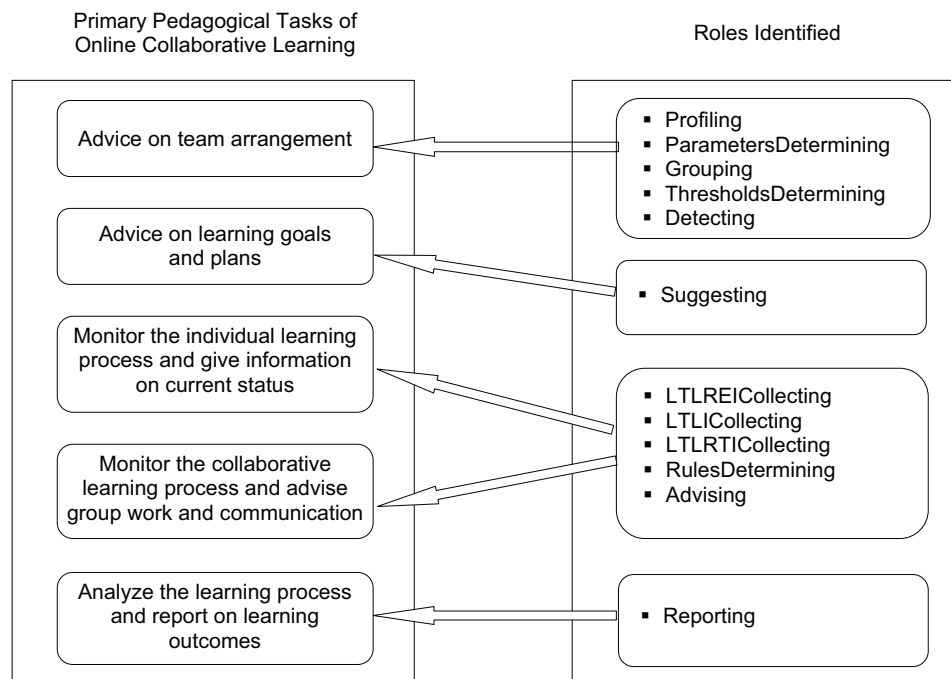
#### **3.1 The environment model**

The environment model is described as a list of computational resources with which roles are associated. Before building the environment model, we name the roles as follows for carrying out the pedagogical tasks for online collaborative learning. The attributes of the roles are further discussed in Section 3.2.

Twelve roles are identified from the general model of online collaborative learning (see Figure 1 for the concrete process):

- *Profiling* — extracts learners' learning styles from learning styles questionnaire.
- *ParametersDetermining* — identifies the values of parameters to use for forming collaborative groups.
- *Grouping* — orders, segments and assigns learners into heterogeneous groups according to their learning styles and the grouping parameters.
- *LTLREICollecting* — collects data about learner to learning resource interaction.
- *ThresholdsDetermining* — decides on the thresholds for determining whether the learners have a high, moderate, or low occurrence of various patterns of behaviour.
- *Detecting* — identifies learners' learning styles from the collected data about learner to learning resource interaction according to specified thresholds.
- *LTLICollecting* — collects data about learner to learner interaction.
- *LTLRTICollecting* — collects data about learner to learning result interaction.
- *RulesDetermining* — determines the rules for formulating collaborative learning advice (e.g. lack of discussion and collaborative knowledge creation, passive learner in discussion, low-level cohesiveness and interaction and so on) on the basis of models of interaction.
- *Advising* — generates collaborative learning advice by analyzing the data (i.e. data about learner to learning resource interaction, data about learner to learner interaction and data about learner to learning result interaction) collected from the learning process on the basis of the rules determined.
- *Reporting* — provides statistical reports on the learning outcomes for individual learners by analyzing the data collected from the learning process.

- *Suggesting* — uses assessment data (e.g. grades received for online quizzes and final transcript information) from passed courses to make suggestions on new learning modules and information resources.



**Figure 1:** Identifying roles in GAOOLE from the primary pedagogical tasks identified in Suh and Lee's model of online collaborative learning (Suh and Lee 2006)

As shown in Figure 2, the environment model consists of thirteen types of computational resources. The middle layer of boxes represents the set of environments that the twelve roles are immersed in; the top and bottom layers of ovals represent the twelve roles specified above; the dashed arrows represent the 'read' relationship between the roles and the associated resources; the solid arrows represent the 'generate' and/or 'modify' relationships between the roles and the associated resources. Take the 'Grouping' role as an example. It reads learners' learning styles and grouping parameters (e.g. the number of learners per group or the number of groups to be formed) to carry out the task of formulating collaborative groups for online collaborative learning activities. The 'Grouping' role also generates collaborative groups resource. A collaborative group is treated as a type of computational resource and each has their own symbolic names, the memberships, and the online lessons that it is involved in.

### 3.2 The roles model

The roles model is used to clarify function and formalize skills (responsibilities, activities) of agents (Huang, et al. 2007). Our roles model contains a collection of twelve tables. Each table specifies the attributes of a single role, namely protocols, activities, permissions and responsibilities, which are used to describe the features and states of agents. Take the 'Grouping' role as an example for showing such a specification table (see Table 1).

In Table 1, the protocols represent the inter-role interactions in which the 'Grouping' role plays a part; the activities represent the private actions that the 'Grouping' role performs; the permissions describe the 'right' associated with the 'Grouping' role; and the responsibilities address the expected behaviour of the role. The responsibilities are specified using Gaia liveness expressions which defines the "life-cycle" of the role. For example, the expression of responsibilities in Table 1 says that 'Grouping' consists of executing the protocol *GetLearningStyles(Requesting)*, followed by the protocol *GetGroupingParameters(Requesting)* and the activity *FormCollaborativeGroups*. The symbol '+' represents the sequential execution of these protocols and activities are then repeated for one or

more times. More information about Gaia liveness expressions can be found in (Zambonelli, et al. 2003).

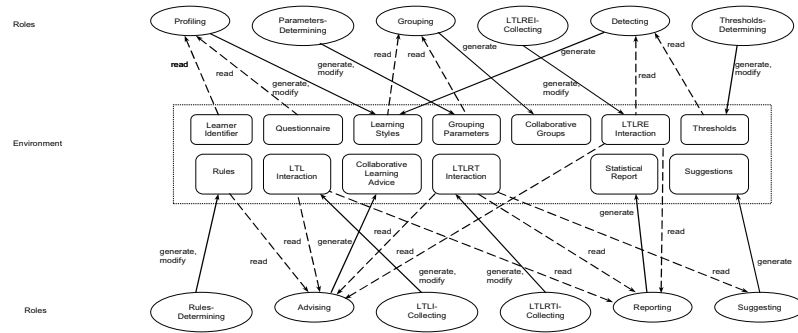


Figure 2: The environment - roles diagram

Figure 2: The environment - roles diagram

Table 1: The roles model - 'Grouping' role

Role Schema:	Grouping
Description:	It asks the 'Profiling' role for learners' learning styles and asks the 'ParametersDetermining' role for the grouping parameters. It formulates collaborative groups for an online collaborative learning activity based on the obtained learners' learning styles and grouping parameters.
Protocols:	GetLearningStyles(Requesting), GetGroupingParameters(Requesting)
Activities:	<u>FormCollaborativeGroups</u>
Permissions:	read learning styles, read grouping parameters, generate collaborative groups
Responsibilities:	Grouping = (GetLearningStyles(Requesting). GetGroupingParameters(Requesting). <u>FormCollaborativeGroups</u> ) <sup>+</sup>

Other roles are specified using the same template as that of Table 1. A short version of the specifications of other roles (contains only the responsibilities) is given in Table 2.

Table 2: The roles model - other roles

Role: Profiling
Responsibilities: Profiling = (GetNewLearner. SendQuestionnaire. GetAnswerInformation. CalculateLearningStyles. StoreResult. (GetLearningStyles(Responding)  UpdateLearningStyles(Responding))) <sup>+</sup>
Role: ParametersDetermining
Responsibilities: ParametersDetermining = (IdentifyGroupingParameters. GetGroupingParameters(Responding)) <sup>+</sup>
Role: LTLREICollecting
Responsibilities: LTLREICollecting = (CollectLTLREIData. GetLTLREIData(Responding)) <sup>+</sup>
Role: Detecting
Responsibilities: Detecting = (GetLTLREIData(Requesting). GetThresholds(Requesting). IdentifyLearningStylesFromLTLREIData. UpdateLearningStyles(Requesting)) <sup>+</sup>
Role: ThresholdsDetermining
Responsibilities: ThresholdsDetermining = (IdentifyThresholds. GetThresholds(Responding)) <sup>+</sup>
Role: RulesDetermining
Responsibilities: RulesDetermining = (IdentifyRules. GetRules(Responding)) <sup>+</sup>
Role: Advising
Responsibilities: Advising = (GetRules(Requesting). GetLTLIData(Requesting). GetLTLRTIData(Requesting).



GetLTLREIDData(Requesting). <u>FormCollaborativeLearningAdvice</u> . <u>SendAdvice</u> ) <sup>+</sup>
Role: Profiling
Role: LTLICollecting
Responsibilities: LTLICollecting = (CollectLTLIData. GetLTLIData(Responding)) <sup>+</sup>
Role: LTLRTICollecting
Responsibilities: LTLRTICollecting = (CollectLTLRTIData. GetLTLRTIData(Responding)) <sup>+</sup>
Role: Reporting
Responsibilities: Reporting = (GetLTLIData(Requesting). GetLTLREIData(Requesting). GetLTLRTIData(Requesting). <u>WriteStatisticalReport</u> . <u>SendReport</u> ) <sup>+</sup>
Role: Suggesting
Responsibilities: Suggesting = (GetLTLRTIData(Requesting). <u>MakeSuggestion</u> . <u>SendSuggestion</u> ) <sup>+</sup>

### 3.3 The interaction model

The interaction model is used to clarify the relationships between roles and to link the interactive agents (Huang, et al. 2007). Such inter-role interactions are defined as Gaia protocols. A Gaia protocol has six attributes, namely protocol name, initiator, partner, inputs, outputs, and description (Zambonelli, et al. 2003). Our interaction model contains eight tables. Each table specifies the six attributes of a single protocol. Using the 'Grouping' role as an example to illustrate such a specification table, there are two protocols in all that the 'Grouping' role is involved in: *GetLearningStyles* (Table 3) and *GetGroupingParameters* (Table 4).

**Table 3:** The interaction model - protocol 'GetLearningStyles'

Protocol Name: GetLearningStyles	
Initiator: Grouping	Partner: Profiling
Inputs: The identifiers of learners who needs to be assigned into groups	Outputs: Learning styles
Description: The 'Grouping' role requests the 'Profiling' role to provide required learners' learning styles when there is a grouping need. The request will be processed by sending the required learning styles to the initiator.	

**Table 4:** The interaction model - protocol 'GetGroupingParameters'

Protocol Name: GetGroupingParameters	
Initiator: Grouping	Partner: ParametersDetermining
Inputs: The identifier of the activity which the grouping process is required for.	Outputs: Grouping parameters
Description: The 'Grouping' role requests the 'ParametersDetermining' role for providing the required grouping parameters. This request will be processed by sending the parameters that are needed.	

In the above tables, the protocol name gives a brief textual description capturing the nature of the interaction; the initiator addresses the role(s) responsible for starting the interaction; the partner presents the responder role(s) with which the initiator interacts; the inputs defines the information used by the protocol initiator while enacting the protocol; the outputs defines information supplied by the protocol responder during interactions; and the description gives a textual description explaining the purpose of the protocol and the processing activities implied in its execution.

The templates of Table 3 and Table 4 are applied for defining the attributes of other protocols. A short version of the specification of other protocols (contains only protocol name and short description) is given in Table 5.

**Table 5:** The interaction model - other protocols

<i>Protocol Name:</i> UpdateLearningStyles
<i>Short Description:</i> The 'Detecting' role requests the 'Profiling' role to update learning styles when it detects any changes of the learners' learning styles.
<i>Protocol Name:</i> GetLTLREIData
<i>Short Description:</i> The 'Detecting' role, the 'Advising' role or the 'Reporting' role requests the 'LTLREICollecting' role to provide data of learner to learning resource interaction.
<i>Protocol Name:</i> GetThresholds
<i>Short Description:</i> The 'Detecting' role requests the 'ThresholdsDetermining' role to provide thresholds
<i>Protocol Name:</i> GetRules
<i>Short Description:</i> The 'Advising' role requests the 'RulesDetermining' role to provide the rules for generating collaborative learning advice.
<i>Protocol Name:</i> GetLTLIData
<i>Short Description:</i> The 'Advising' role or the 'Reporting' role requests the 'LTLICollecting' role to provide data of learner to learner interaction.
<i>Protocol Name:</i> GetLTLRTIData
<i>Short Description:</i> The 'Advising' role, the 'Reporting' role or the 'Suggesting' role requests the 'LTLRTICollecting' role to provide data of learner to learning result interaction.

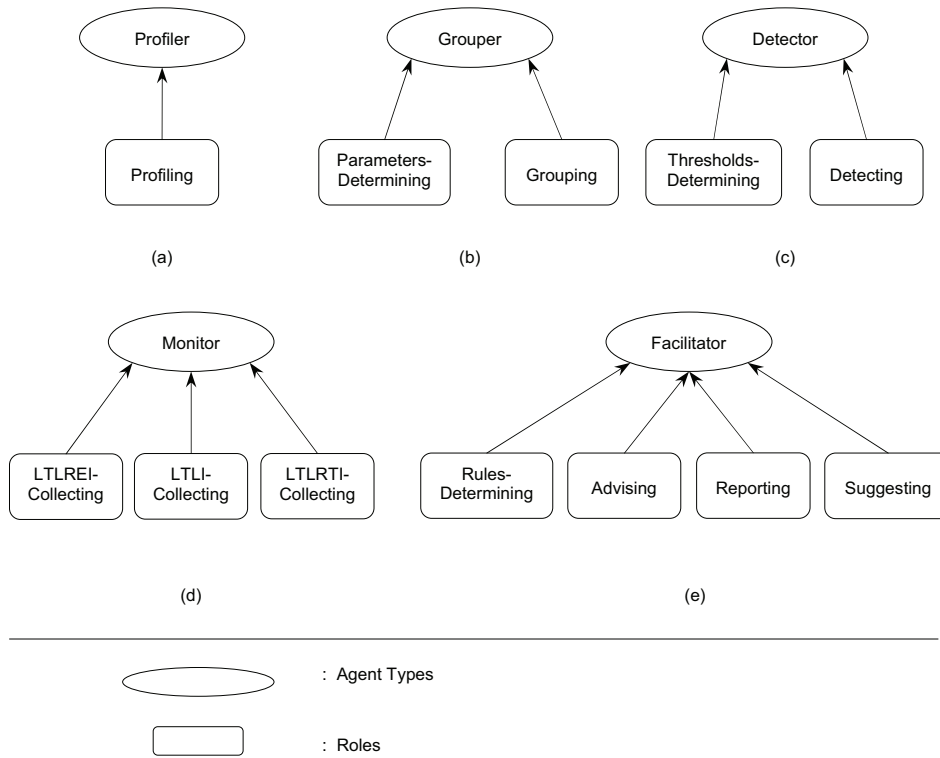
### 3.4 The agent model

The agent model is used to specify the types of agents and the number of instances of each agent type in the actual system. In the Gaia context, an agent is an active software entity playing a set of roles. Thus, the definition of the agent model amounts to identifying which agent classes are to be defined to play specific roles and how many instances of each class have to be instantiated. Typically, there may be a one-to-one correspondence between roles and agent classes. Other considerations when doing the mapping include: coherence of an agent class (how easily its functionality can be understood), efficiency, and respecting to organizational rules (Zambonelli, et al. 2003).

The agent model for our system includes five agent types: the Profiler, the Grouper, the Detector, the Monitor and the Facilitator (see Figure 3). Each of the agent types is assigned one or more roles. For example, the Profiler Agent Type is assigned the 'Profiling' role and the Monitor Agent Type is assigned the roles of 'LTLREICollecting', 'LTLICollecting' and 'LTLRTICollecting'. In addition, the number of instances of each agent type in the actual system is not addressed in this paper since this beyond our focus of study.

In Figure 3(a), the 'Profiling' role extracts characteristics from learners themselves, rather from the interaction. Thus it has different types of activities, permissions and responsibilities with roles such as 'LTLREICollecting', 'LTLICollecting' and 'LTLRTICollecting'. In Figure 3(b), 3(c) and 3(e), to combine one of the roles for setting criteria (e.g. the 'ParametersDetermining' role) with one of the roles that is for the same sub-process of online collaborative learning (e.g., the 'Grouping' role) increases the efficiency of the whole organization since this choice decreases the interactions between different agent classes and cuts the computational cost of adding another agent type into the system (the one which compacts the three roles for setting criteria, i.e. 'ParametersDetermining', 'ThresholdsDetermining' and 'RulesDetermining'). In Figure 3(d) and 3(e), these three roles (for 3(e) excluding the 'RulesDetermining' role) are closely related as they have similar functionality and similar type of protocols, activities, permissions and responsibilities. The choice to aggregate the three roles for one agent type may compact the whole design by reducing the number of agent classes and instances and it may also minimize the conceptual complexity.





**Figure 3:** The agent model

### 3.5 The services model

The services model further specifies the services provided by individual agents. A service can be thought of as a single, coherent block of activity in which an agent will be engaged.

The identification of the services model requires determining the amount of services associated with a single agent type. One or more services can be identified to compose a single agent in that there will be at least one service for each parallel activity of execution that the agent has to execute and there may be a need to introduce more services to represent different phases of the agent execution.

We derive the services for the six agent types from the list of protocols, activities, responsibilities of the roles it implements, which are addressed as follows:

- Profiler Agent: extract learning styles from questionnaire, send learning styles, and update learning styles.
- Grouper Agent: identify the grouping parameters and formulate collaborative groups.
- Detector Agent: identify the thresholds, identify learning styles from interaction data and provide detected learning styles.
- Monitor Agent: collect LTLREI data, collect LTLRTI data, collect LTLI data and provide collected data.
- Facilitator Agent: identify the rules, give advice, give reports and make suggestions.

For each service the properties of the *inputs*, *outputs*, *pre-conditions* and *post-conditions* must be identified (Zambonelli, et al. 2003). Inputs and outputs to services are derived in an obvious way from both the interaction model (for services involving the elaboration of data and knowledge exchange between agents) and the environment model (for services involving the evaluation and modification of environmental resources); pre- and post-conditions represent constraints and states on the execution and completion of services.

Taking the 'Formulate Collaborative Groups' service as an example illustrates the definition of the properties for a single service (see Table 6).

**Table 6:** The services model - 'Formulate Collaborative Groups' service

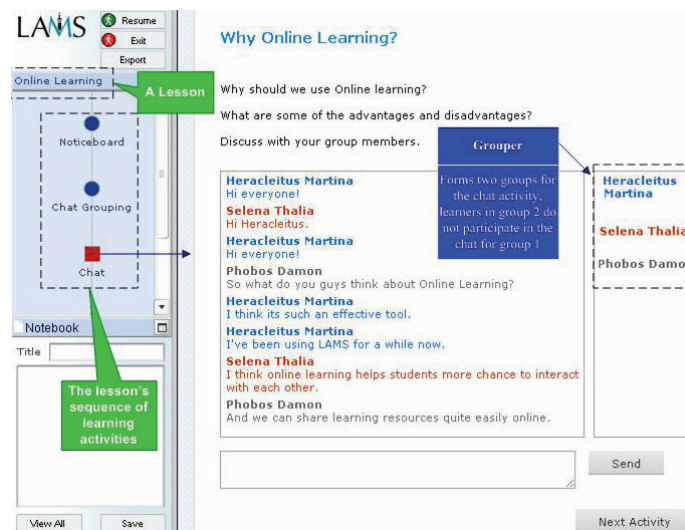
Service	Inputs	Outputs	Pre-condition	Post-condition
Formulate Collaborative Groups	<i>the learning styles of learners who needs to be assigned into groups, the grouping parameter for the collaborative learning activity</i>	<i>the formed collaborative groups</i>	<i>the learners involved in a collaborative learning activity begin the activity through online learning system</i>	<i>the learners involved are assigned into different groups according to the results of the grouping process</i>

The inputs are derived from the outputs of the protocols of 'GetLearningStyles' and 'GetGroupingParameters' (see Table 3 and Table 4). The outputs of this service are derived from the computational resource 'Collaborative Groups' associated with the 'Grouping' role (see Figure 1). The pre-condition represents the constraint of 'the grouping process cannot start unless the learners involved begin the activity' for the execution of the service and the post-condition represents the system state 'all learners involved are assigned into groups' on the completion of the service.

#### 4. Examples of online collaborative learning activities and GAOOLE agent supports

To illustrate the models identified using the Gaia methodology, we choose LAMS (LAMS International 2002) (because of its popularity) as the learning platform to show two examples (Figure 4 and Figure 5) of online collaborative learning activities and the support that can be provided by GAOOLE agents in implementing these models.

In Figure 4, learners can be divided into several groups and participate in the online 'chat' activity. This functionality can be supported by a Grouper agent in GAOOLE.



**Figure 4:** A screenshot of LAMS (LAMS International 2002) 'chat' activity for a lesson 'Online Learning' and the support from a Grouper agent

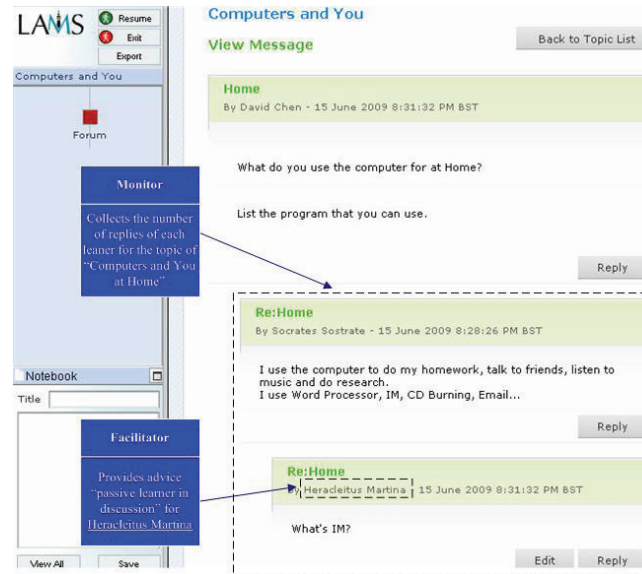
In Figure 5, the number of replies of each learner to a topic posted in a 'forum' activity can be collected and used to generate the learning advice. These functionalities can be supported separately by a Monitor agent and a Facilitator agent in GAOOLE.

#### 5. Integration of GAOOLE agents in a Collaborative Learning Environment

In order to demonstrate the flexibility and pro-activeness of GAOOLE, we present the architecture of integration of GAOOLE agents in a CLE (see Figure 6). The main concept behind this architecture is to use a multi-agent platform to design and implement the features (as designed as GAOOLE agents) that will provide the added value to existing collaborative learning environments. The basic aspect of the architecture is the fact that the agent platform communicates with the collaborative learning

environment through the use of web services, more specifically by using the SOAP protocol. The architecture has the following components:

- *Web Client*: consisting of a browser interacting with the collaborative learning environment.

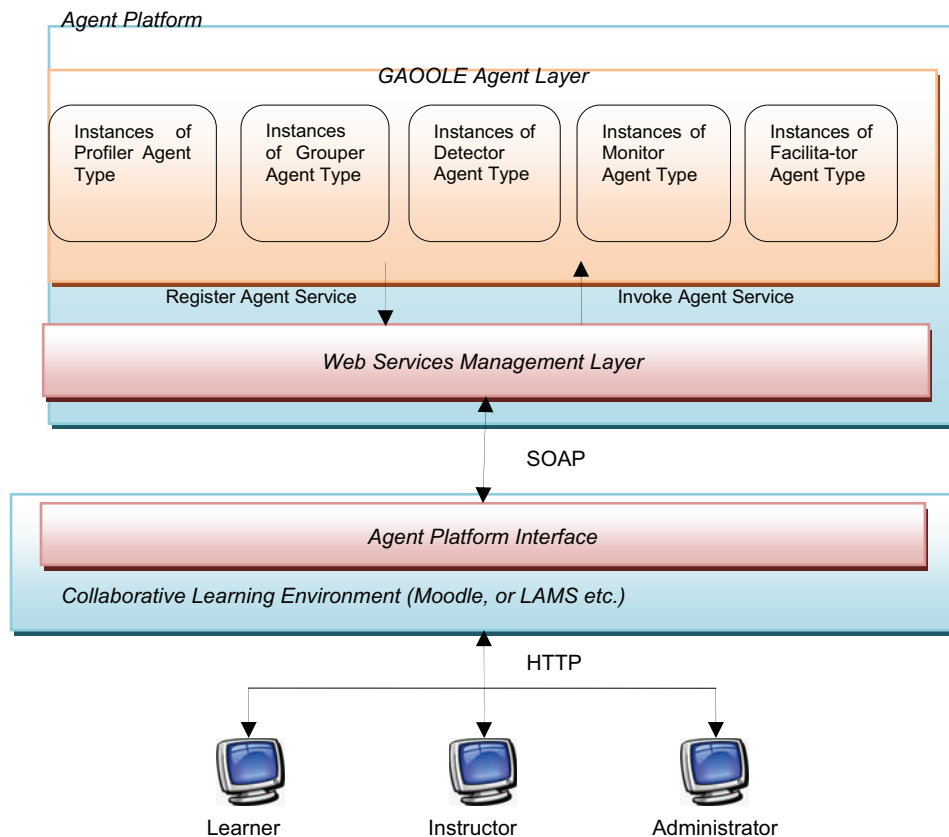


**Figure 5:** A screenshot of LAMS (LAMS International 2002) 'forum' activity for a lesson of 'Computers and You' and the support from a Monitor agent and a Facilitator agent

- *Collaborative Learning Environment*: consisting of an exiting collaborative learning environment such as Moodle, LAMS etc. that provides a multi-functional working area for users to carry out teaching and learning related activities. In Figure 6, there is an *Agent Platform Interface* within the collaborative learning environment to send SOAP request messages to and receive SOAP response messages from the agent platform.
- *Agent Platform*: consisting of a *Web Services Management Layer* and a *GAOOLE Agent Layer*. The former layer deals with issues of registering agent services as web service endpoints and the invocation of agent services for requests from web service clients. The SOAP requests coming from the collaborative learning environment are processed by the *Web Services Management Layer* and allocated to the appropriate agent in order to be processed. The SOAP responses are sent back to the *Collaborative Learning Environment*. The *GAOOLE Agent Layer* comprises a collection of instances of agent types as specified in the proposed *GAOOLE* models. These agents provide collaborative learning specific functionalities as described in Section 1.

In order to make use of this integration, it may be necessary for some collaborative learning environments to add an *Agent Platform Interface* in order to cooperate with the *Agent Platform*. Moreover, there is a variety of choices to implement the agent platform, such as the Java Agent DEvelopment framework (JADE) (Bellifemine, et al. 2003) and the Tryllian Agent Development Kit (ADK) (Tryllian 2003). However, not every agent platform supports the use of web services. Thus, it is also desirable to add a *Web Services Management Layer* to the selected agent platform if it does not support web services. For example, the Jade Web Services Integration Gateway (WSIG) add-on (JADE Board 2005) plays the role of web services management as required in Figure 6.

The integration of *GAOOLE* agents in conventional collaborative learning environments such as Moodle, Blackboard and others, contributes to the research area of intelligent collaborative learning environment through extending the functionality of them by adding modules of structuring and managing the collaboration such as organizing teams, advising on group work and communication, tutoring, and testing individual contributions.



**Figure 6:** Architecture of integrating GAOOLE agents in a CLE

## 6. Conclusions and future work

In this paper, we present the design analysis of a CLE using an established agent-based methodology. Our approach provides the functionalities that are often missing in existing collaborative learning environments, including grouping learners based on their characteristics, detecting learners' learning styles, collecting interaction data, constructing model of interaction, comparing with desired state, and moderating. Applying an agent-based methodology will enable the strengths of agents to be exploited. Specifically, it enables the system to be pro-active, flexible and efficient while enabling the system to be implemented and integrated with existing components and systems.

A prototype of GAOOLE is under development, and will enable us to validate the Gaia models, and refine them where appropriate. Future work also includes an evaluation of the system effectiveness and efficiency.

## References

- Anderson, A., Cheyne, W., Foot, H., Howe, C., Low, J. and Tolmie, A. (2000) "Computer Support for Peer-based Methodology Tutorials", *Journal of Computer Assisted Learning*, Vol 16, No. 1, pp. 41-53.
- Baylor, A.L. (2002) "Agent-based Learning Environments as a Research Tool for Investigating Teaching and Learning", *Journal of Educational Computing Research*, Vol 26, No. 3, pp. 227-248.
- Bellifemine, F., Caire, G., Poggi, A. and Rimassa, G. (2003) "JADE: A White Paper", *EXP in search of innovation (Special Issue on JADE)*, Vol 3, No. 3, pp. 6-19.
- Blackboard Inc. (1997) "Blackboard Home", [online], Blackboard Inc., [www.blackboard.com](http://www.blackboard.com). (Accessed 17 May 2009).
- Cohen, E.G. (1994) "Restructuring the Classroom: Conditions for Productive Small Groups", *Review of Educational Research*, Vol 64, No. 1, pp. 1-35.
- Hamdi, M.S. (2007) "MASACAD: A Multi-Agent Approach to Information Customization for the Purpose of Academic Advising of Students", *Applied Soft Computing*, Vol 7, No. 3, pp. 746-771.
- Huang, W., El-Darzi, E. and Jin, L. (2007) "Extending the Gaia Methodology for the Design and Development of Agent-based Software Systems", *Proceedings of the 31st IEEE Annual International Computer Software and Applications Conference (COMPSAC 2007)*, Beijing, China, pp. 159-168.

- JADE Board (2005) "Jade Web Services Integration Gateway (WSIG) Guide", [online], Whitestein Technologies AG, [http://jade.cse.it/doc/tutorials/WSIG\\_Guide.pdf](http://jade.cse.it/doc/tutorials/WSIG_Guide.pdf). (Accessed 18 May 2009).
- Kosba, E. (2004) *Generating Computer-based Advice in Web-based Distance Education Environments*, Doctor of Philosophy Thesis, School of Computing, University of Leeds.
- LAMS International (2002) "LAMS: Learning Activity Management System", [online], LAMS International, <http://www.lamsinternational.com/>. (Accessed 17 May 2009).
- Lin, F., Esmahi, L. and Poon, L. (2005) Integrating Agents and Web Services into Adaptive Distributed Learning Environments, In: *Designing Distributed Learning Environments with Intelligent Software Agents*, Idea Group Inc (IGI), pp. 184-217.
- Liu, Z., Jin, H. and Fang, Z. (2006) "Collaborative Learning in E-Learning based on Multi-Agent Systems", Proceedings of the 10th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD 2006), Nanjing, China, pp. 1-5.
- Mencke, S. and Dumke, R. (2007) "A Framework for Agent-Supported e-Learning", Proceedings of the International Conference on Interactive Computer Aided Learning (ICL 2007), Villach, Austria.
- Moodle.org (2005) "Moodle: Open-Source Community-based Tools for Learning", [online], Moodle Pty Ltd, <http://moodle.org/>. (Accessed 17 May 2009).
- Piramuthu, S. (2005) "Knowledge-Based Web-Enabled Agents and Intelligent Tutoring Systems", *IEEE Transactions on Education*, Vol 48, No. 4, pp. 750-756.
- Razek, M.A., Frasson, C. and Kaltenbach, M. (2002) "Toward More Cooperative Intelligent Distance Learning Environments", Proceedings of the Workshop on Software Agents - Co-operation - Human Activity (SACHA 2002) - ITS Conference, Biarritz, France.
- Reimann, P. (2003) "How to Support Groups in Learning: More than Problem Solving. Invited Keynote", Supplementary Proceedings of the 11th International Conference on Artificial Intelligence in Education (AIED 2003), Sydney, Australia, pp. 3-16.
- Soh, L.-K., Khandaker, N., Liu, X. and Jiang, H. (2006) "A Computer-Supported Cooperative Learning System with Multiagent Intelligence", Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, pp. 1556 - 1563.
- Suh, H. and Lee, S. (2006) "Collaborative Learning Agent for Promoting Group Interaction", *ETRI*, Vol 28, No. 4, pp. 461-474.
- Tryllian, B. (2003) "Tryllian Agent Development Kit", [online], Tryllian BV, <http://www.tryllian.org/>. (Accessed 18 May 2009).
- Wooldridge, M., Jennings, N.R. and Kinny, D. (2000) "The Gaia Methodology for Agent-Oriented Analysis and Design", *Journal of Autonomous Agents and Multi-Agent Systems*, Vol 3, No. 3, pp. 285-312.
- Zambonelli, F., Jennings, N.R. and Wooldridge, M. (2003) "Developing Multiagent Systems: The Gaia methodology", *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Vol 12, No. 3, pp. 317-370.
- Zumbach, J., Reimann, P. and Koch, S.C. (2006) "Monitoring Students' Collaboration in Computer-Mediated Collaborative Problem-Solving: Applied Feedback Approaches", *Journal of Educational Computing Research*, Vol 35, No. 4, pp. 399-424.